An SE Approach for CoCo Learning of Virtual Labs

Lalit Mohan Sanagavarapu
Software Engineering Research Centre
IIIT Hyderabad
Hyderabad, India
lalit.mohan@research.iiit.ac.in

Mrudhivka Damaraju
Virtual Labs
IIIT Hyderabad
Hyderabad, India
mrudhvika@vlabs.ac.in

Ravi Shankar Pillutla
Virtual Labs
IIIT Hyderabad
Hyderabad, India
ravi@vlabs.ac.in

Venkatesh Choppella
Software Engineering Research Centre
IIIT Hyderabad
Hyderabad, India
venkatesh.choppella@iiit.ac.in

Y R Reddy
Software Engineering Research Centre
IIIT Hyderabad
Hyderabad, India
raghu.reddy@iiit.ac.in

Priya Raman
Virtual Labs
IIIT Hyderabad
Hyderabad, India
priya@vlabs.ac.in

***Abstract*—With increasing computing power, capacity and internet connectivity, the ability to learn has crossed classroom boundaries in the form of online learning. The adoption rate and effectiveness of online learning continues to be an area of research. Collaborative and Cooperative (CoCo) learning involves a joint effort by students to understand and solve problems or tasks. Adoption of open source software has also increased with technology-aided collaboration and cooperation. Leveraging the practices of crowdsourced software development, we propose CoCo Learning approach for technical education using virtual labs. About 145 undergraduate engineering students from various institutes participated in our study for a period of one full semester. About 92% of the students conducted an average of 24 virtual experiments with this approach. Nearly 90% of the faculty from these institutes also confirmed that their students' understanding of courses has improved while delivering tangible software artifacts as outcomes.**

***Index Terms*—Collaborative Learning, Cooperative Learning, Virtual Labs, Software Engineering**

I. Introduction

Traditional learning is more like the 'sage on the stage' approach in which the teacher directly interacts with students in a physical location and gives students sources for learning. To a certain extent, it restricts students from exploring new concepts by themselves. Flipped classroom, scavenger hunting, problem solving and other blended methods are being implemented to enhance student interaction and engagement [1]. The core of an "effective learning approach" rests in the ability of a person to absorb and internalize, and recall whenever is required [2]. With increased access of internet, computer supported learning in the form of online books, videos, Massive Open Online Courses (MOOCs), Virtual Labs, Virtual Reality assisted learning and other forms has provided scalability and availability of best resources to students [3].

Despite heavy technological investments and deeper research in computer supported learning, they have opened newer challenges on student engagement and learning. Some of the available statistics state that the drop rate in online learning is more than 90% due to lack of support, failure in understanding the content, usability, effectiveness and other reasons including lack of interpersonal interactions [4] [5]. Exploring and interacting with other students while learning is identified as active and constructive learning [6]. We hypothesize that an approach that supports exploration and interaction such as cooperative and collaborative learning would further strengthen online learning.

The advantages such as better remembrance, accountability, quality of interpersonal interactions, self-esteem from active learning approaches such as Collaborative and Cooperative learning are well established [7] [8]. Collaborative and Cooperative learning are interactive learning approaches that require social and intellectual effort from multiple students. Collaborative learning demands responsibility, persistence and sensitivity amongst the students leading to a community of learners encouraging others to join, participate, grow [9] and share their learnings in a community [10]. Cooperative learning requires students to work together in small groups to solve problems [11]. Some of the key differences between cooperative and collaborative learning are well described by Olivers [12].

With cooperative and collaborative learning approaches, students are expected to not just read new information but are also expected to build something new with the processed information. Computer-supported collaborative or cooperative learning (CSCL) addresses some of these challenges [13]. CSCL requires focus on the meaning-making practices of collaborating groups and on the design of technological artifacts to mediate interaction with focus on individual learning. Performing activities such as planning, organizing and commenting have become easier with CSCL. However, creating a collaborative learning experience, achieving consensus and grading students, continues to be some of the challenges. Based on our literature study, most of the current CSCL research has been to create content, integrate LMS into e-Learning and develop recommenders to address learning challenges [14].

978-1-7281-6807-4/20/$31.00 ©2020 IEEE

We focused on scalable simulation based virtual lab [15] for implementing and measuring effectiveness of learning in collaborative and cooperative approach. Virtual labs can massively scale and reach various locations of the world enabling teachers and students to collaborate. More specifically, virtual labs aid institutions that have scarce resources; students in these institutions are deprived of learning due to lack of lab infrastructure. These labs provide instant feedback and offer flexibility to redo the procedure steps enhancing critical thinking. The simulation software repositories of virtual labs contain reusable assets, an ongoing list of issues and enhancements for future releases. To overcome the limitations of collaborative and cooperative learnings independently, we propose a systematic combination of these approaches that lead to effective learning over a longer period, in which students can fix and enhance virtual labs software by programming while conducting experiments or performing related activities.

Software programming is not specifically to computer science students but is considered as 'twenty-first century literacy' [16]. Since there are equally good programmers from non-computer science educational background, programming is now a tool that people use as part of their learning to perform day-to-day jobs. Numerous online videos, study material and environments to practice programming are available such as CodeChef, Coursera and others. Hence, we theorize that fixing or developing Virtual Labs software is related with the understanding of the discipline, domain, course or subject rather than just programming. Further, developing, fixing and validating Virtual Labs while learning can be considered as crowdsourcing of software engineering (SE).

We reviewed the literature on crowdsourcing of software development [17] to understand the challenges and motivation in programming. Intrinsically or extrinsically motivated [18] workers (programmers and testers) are contributing to open source software in a large number. Task assignment and validation have been an area of focus in crowdsourcing software development [19], [20]. The cooperative practices of agile methodologies including daily stand-up meetings to discuss work items 'completed', 'to be completed' on that day and 'impediments' increased the deliverable transparency [21] in software development. Based on the identified advantages of cooperative and collaborative learning, success of virtual labs in education, standardised SE practices, and growth in software crowdsourcing with enhanced collaboration tools, we propose a Software Engineering approach for CoCo (Collaborative and Cooperative) Learning in virtual labs. In the further sections of the paper, we detailed our proposed approach, discussed our study with Virtual Labs, analyzed results based on empirical studies, concluded with our observations and the potential future work.

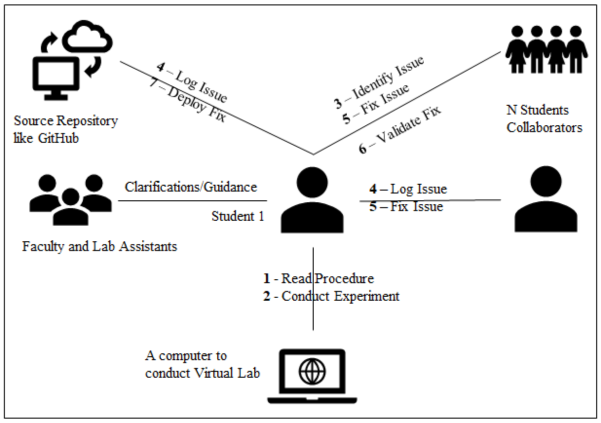

Fig. 1. Process of CoCo Learning

II. Approach for CoCo Learning

Collaborative learning requires students to work on a common task together interacting actively whereas in cooperative learning students work on individual tasks towards a common problem along with the mentor. We combine these educational approaches as CoCo Learning to obtain a synergistic result in higher education such as Engineering, and technology oriented courses that require problem solving and experimentation that is beyond memorizing, remembering and understanding [22]. We propose to use points grading mechanism for various activities in CoCo learning with virtual labs so that it aligns with institutes' grading methods. The Figure 1 shows the process and the interaction between students from various institutes and other stakeholders. The steps in our CoCo Learning approach on virtual labs are

- Students read theory, objective and procedure to conduct a lab experiment on their own and analyze results. They could read in a group and seek clarification on collaboration tools to understand the procedure or validate the results. Typically, exploring or using the experiment is the last step in a virtual environment.
- In the proposed approach, SE practices of issue identification, impact analysis, coding and testing on virtual lab experiments are adopted. Students collaborate to obtain common understanding of the lab to identify and log a valid issue.
- A valid issue (tagged as an 'open-issue') is logged in the source code repository of virtual labs. To log an issue, students practice SE tasks related to documenting defects or issues. Each issue contains 'Issue Name', 'Description', 'Steps to re-produce issue', 'Severity' and 'Expected fix'. As an issue is logged by a student, other students are expected to cooperate in documentation as they collaborated in identifying an issue. To engage students, a collaboration tool is proposed that poses three questions everyday seeking information on (i) tasks completed in the previous day (ii) tasks to be completed on that day (iii) tasks that require assistance.

These questions are similar to daily stand-up meetings in Agile Scrum software development methodology.

- Duplicate logged issues are tagged as 'Dup-Issue' so that students get to fix only unique issues. By reviewing the available literature on duplicate issues in GitHub [23], a utility for tagging duplicate issues is utilized.
- A list of open unique issues are available to students for fixes. Once the issue is fixed, students send a request to validate the issue on the collaboration tool (tagged as 'ValidateIssue'), or students query to get a list of issues available for validation. As there are many students planning to work on a fix, they collaborate or cooperate while working on a fix to avoid duplicate effort.
- Caper Jones [24] prepared a heuristics based mapper on lines of code to a Function Point (FP) per programming language. Points can be less than 1 as well based on the number of lines of code programmed [25] for completing the task. The 'compare' and 'cloc-git' features in GitHub are leveraged to get the lines of code changed as part of a fix and use a mapper to calculate FP. We understand Caper Jones table does not have scientific rigour, our intent is not software estimation but only to measure the activities in CoCo Learning. Students validate fixes and get points awarded. They are also given points to identify invalid issues. Students fixing issues, use the collaboration tool to (1) seek clarification on a issue, (2) get programming language support, and (3) validate fixes. Students who assist other students on collaboration platform are awarded with additional 0.1 points for every assistance that is acknowledged by a requestor.
- Students who are not comfortable with programming are awarded points based on the count of validations performed on fixed issues. To create equal opportunities of learning and improve collaboration, two students are required to validate every fix to reduce the scope of error in validation. Only one validation is allowed from a student belonging to the same institute. The second validation has to be done by a student from different institute. This resolves concerns of quality and allows collaboration amongst students across institutes. Students validate a fixed issue (tagged as 'validatedfix') and capture a screen shot of the results and log it against the issue in the repository.
- Validated fixes are reviewed for deployment by faculty or lab teaching assistants involved in developing labs. Unapproved fixes are re-opened for fixes (tagged as 'open-issue'). Students who have validated rejected fixes are penalized (-1 multiplied by number of FPs) to ensure fairness in the process. Students are also awarded for logging the issues, the weightage for identification of an issue in a experiment simulation is more as compared to theory, quiz, etc available in virtual lab. This proposed award and penalty points mechanism reduces quality concerns in crowdsourcing [26].

The duration of the proposed approach is for 4-6 months (semester) so that learning is paced over a sustainable period and the grading can be used in course evaluation as well. The Table I distinguishes collaboration and cooperation activities while conducting virtual labs. We propose that faculty or lab assistants to be available as moderators on collaboration tool to clarify questions that are not answered by any student.

Activities	Individual	Cooperate	Collaborate
Read Procedure	✓		
Conduct Experiments	✓		
Identify Issues			✓
Log Issues		✓	
Fix Issues		✓	✓
Validate Fix			✓
Deploy Fix	✓		

TABLE I
ACTIVITIES IN COCO LEARNING

III. IMPLEMENTATION OF COCO LEARNING

Some of the widely used virtual labs are eLabFTW[51], Virtual Labs from Government of India [27], Virtual Hazards Impact and Risk Laboratory and KiCad EDA. GitHub, GitLab, BitBucket and others are some of the open source repositories which contain source code of virtual experiments and the simulators. Slack, RocketChat, Jive and other forums are some of the popular collaboration tools.

For validating our proposed approach, we used Virtual Labs, an initiative of Ministry of Human Resource Development from Government of India with approximate 4 million usages of 114+ labs across nine engineering and two science disciplines. Unlike other available platforms on web, Virtual Labs has a wide range of experiments across engineering disciplines. The labs were built by faculty and students of some of the top engineering institutes from India to augment learning in Tier 2 and 3 institutes, rural students who may not have access to the necessary lab infrastructure. The source code for Virtual Labs is available on GitHub [28]. The repository has more than a million lines of code and 400+ contributors from different parts of the world. Sophomore and junior Computer Science and Engineering (CSE) and Non-CSE students from eleven Tier 2 and 3 engineering institutes have participated in our study. The faculty and the students were not informed about the intent of the study to avoid bias. We provided students a period of one month to identify issues in the 24 Virtual Labs. We selected Slack[2], a widely accepted collaboration tool in the industry to reduce the onboarding time. After the completion of all activities that included closing fixed issues and migrating changes to production, an online survey was conducted with students and the faculty to validate the effectiveness of the proposed approach. The details of the questions for students' survey are given in Table II.

[1] https://www.elabftw.net/

[2] https://vlabsoutreachinterns.slack.com/

SQ#	Survey Question	Purpose
1-4	Student name, Institute name, Engineering discipline, Year of study	To bring seriousness to the survey and validity to the data for any future reference
5	How many labs (a rough count) have you tested to identify issues?	Insight on the exploration and usage of Virtual Labs
6	How many lab experiments (a rough count) have you fixed or resolved defects?	To obtain information of students who would have fixed issues but not raised a pull request in the software repository of the lab
7	How often have you interacted with students from your or other institutes to complete the task?	To obtain an insight on the need for cooperation and collaboration amongst students.
8	How much has your understanding of labs improved because of testing, fixing and interacting with other students?	Insight on the effectiveness of our approach
9	What did you like the most in these activities?	Textual response to capture comments on activities
10	What could have been better to motivate students for active contribution?	Textual response to understand other concerns or inputs on the approach.

TABLE II
STUDENT SURVEY QUESTIONS

As CoCo learning was being tried for the first time, we did not involve participating institute faculty on the collaboration platform. Authors performed the role of Lab Assistants and Faculty on Slack. However, faculty from participating institutes coordinated with us to provide students for the study to understand the importance of Virtual Labs. A survey with 5 questions were conducted with those faculty to seek their inputs, the survey questions were -

- **FQ1-FQ2** - Name of the faculty and institute name. The response to this question was used to check whether the responses were from faculty who had students participation in CoCo learning and the survey.
- **FQ3** - What is the scale of improvement in the students' understanding of virtual labs because of participating in fixing and validating activities? The provided option's were 'No Improvement', 'Not Much', 'Average', 'High', 'Very High'. The response to this question could be correlated with the responses to the question SQ8 from students.
- **FQ4** - What did you like in this mode of Virtual Labs participation? The responses to this question was in plain text and provides an insight on the scalability of the learning approach.
- **FQ5** - What could have been better for improved contribution? The response to this question was in plain text and provides an understanding on areas of improvement to make learning effective.

IV. Results and Analysis

A request was sent to 21 engineering colleges to nominate 10 students per college for 6 months to participate in fixing and developing Virtual Labs. About 145 students signed up on Slack. The students were given a month time to read the procedures, conduct virtual lab experiments and log issues on Virtual Labs GitHub [29]. Students logged 1,026 issues including 95 as enhancements. Most of the logged issues were related to broken links, missing or incomplete content, minor functionality issues on logic and improper programming practices. The enhancements issues related to feature or simulation as 'functionality' issues, Defects such as broken links, improper programming and others as 'Operational' issues. We observed that more issues were logged during the beginning of study and the velocity of issue logging reduced as the time progressed. The issue (I) count reduced from 1,026 to 545 after removal of duplicates using an internally developed utility. The Table III shows the types of issues that were identified by the students, number of enhancements and features suggested during the period.

Issue Type	CSE	Non-CSE	Sciences	Total
Functionality Issues	108	47	29	184
Broken Links and Code Clean-up	132	81	53	266
Enhancements and New Features	58	24	13	95
Total	298	152	95	545

TABLE III
TYPE OF ISSUES IDENTIFIED BY STUDENTS

As shown in Table IV, 65% of I, i.e., 355 fixes (F) were fixed by students in the remaining 5 months. Here again, the count of issues being fixed reduced as we got closer to the end of the period. From the initial count of 145 students, 11 students did not show any active participation due to conflicting schedule with their course work and health conditions. Approximately, 79% of F fixes, i.e., 280 issues were validated by 2 students and 21% of F, i.e., 77 issues were validated by one student confirming that collaboration is possible in virtual lab environment for technology courses.

Fix Type	CSE	Non-CSE	Sciences	Total
Functionality	61	39	19	119
Operational	113	74	49	236
Total	174	113	68	355
% of Fix	58%	74%	72%	65%

TABLE IV
TYPE OF FIXES BY STUDENTS

The figure 2 shows the trend of fixing functional and operational issues across labs. The trend states that students took more time to fix functionality issues as compared to operational issues. The reason could be that operational issues (89% are fixed) were technologically easy to fix or that the students required more time to fix or enhance a functional issue (43% are fixed). The trend displaying the fixing of the functional issue shows that the students were able to understand the lab functionality. It was also observed that more number of functionality issues in CSE labs were fixed. The scored points of students are available on the GitHub page [30]. This confirmed that students contributed to fixes and enhancements of Virtual Labs. Also, some of them were more motivated to perform in an online environment. Although the count of fixed issues of CSE related labs were more, the relative percentage of issues fixed in Non-CSE labs is high. This states that the Non-CSE students were able to program irrespective of their engineering discipline.

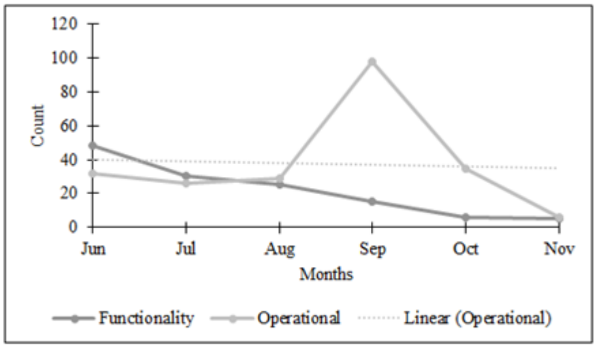

Fig. 2. Trend of Operational and Functional Issues

With the available data on Slack, we observed that 44,812 messages were exchanged either on public forum or messaged directly. About 64% of the students read messages posted, 12% of the students posted on public forum and 88% posted direct messages. We can infer from the Slack posts that students have learned to use collaboration tools for interaction to understand and fix issues. The interactions were also due to the Slack bot seeking daily progress, collaboration for validations and clarifications on the issues.

To further validate the effectiveness of the approach, we conducted online survey[3] with students and faculty. About 68% of the students participated in the survey. The ratio of CSE to Non-CSE students were almost equal, 58% of the students were junior year undergraduates. In response to the survey question **SQ5**, it was observed that the count of experiments conducted in a virtual environment by the students were more than physical lab experiments performed as part of their regular lab work during a semester.

From the response to **SQ6**, it was observed that students fixed issues in approximately 16 labs. Sophomores were twice (count of fixes) more enthusiastic as compared to juniors. The reason could be because junior year students might not have performed the physical labs as part of their semester work and their lack of agility in programming or engineering a solution. There was a strong correlation (Spearman correlation score - 0.86) between count of lab experiments tested vis-a-vis count of issues fixed. As shown in Figure 3 in response to survey question **SQ7**, 77% students actively collaborated or cooperated with students from other institutes to complete the given task. The figure also shows the extent of interaction vis-a-vis % of students. Interestingly, there was no visible difference in the improvement rating of juniors and sophomores. This implied that both groups agreed that Virtual Labs had helped them to understand the course. The results to **SQ8** survey question pointed 92% of the students have stated that their understanding of lab experiments have improved after taking part in testing, fixing and other collaboration activities. The spearman correlation score between **SQ7** and **SQ8** is 0.52, statistically supporting that interactions helped in improving the understanding of the course. Conceptual understanding, problem solving, and collaboration were the broad responses to **SQ9** survey question on liking in the approach. Publishing leader boards and conducting a workshop/bootcamp for understanding the technology stack were the main themes to **SQ10** on improvements in the approach.

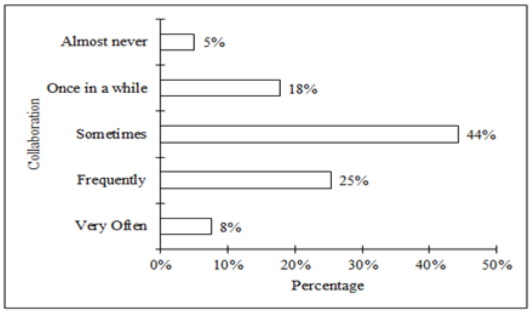

Fig. 3. Frequency of Collaboration

Twenty faculty members from the participated institutes in the CoCo Learning responded to the online survey except one. In response to **FQ3**, faculty stated that student learning improved by 90% in the courses after participating in Virtual Labs related activities. This percentage correlates with the response provided by students to **SQ8** survey question. Both groups confirmed that understanding of lab experiments had improved course understanding, validating the usefulness of the CoCo learning approach. The broad answers from **FQ4** were 'flexibility to learn anytime and anywhere' and 'easy visualization of practical aspects'. In response to the survey question **FQ5**, faculty requested more virtual labs, better alignment to their respective college syllabus and improvements to the user interface of the labs.

With CoCo Learning approach, students developed the ability to get into details to find an issue, understand the source code developer by other developers to fix issues, work on GitHub and, software linting tools. Students

[3] https://forms.gle/HCA1N9eBJGjiwhg96

interacted and collaborated with students from different institutes that enabled them to come out of comfort zone and build social skills. As observed from the complexity of fixed issues that the students fixed, their Problem solving skills apart from IT skills had also improved. With Virtual Labs being one of the largest open source repositories for computer supported learning, contributing on GitHub enhanced their resume. As the practices followed were similar to Agile software development methodology of issue logging and daily stand-up meetings, students may also perform better in job interviews and present their readiness to work in IT and lean manufacturing organizations.

V. Conclusions and Future Work

We analyzed and discussed the implementation of the cooperative and collaborative learning approach. Students from different institutes collaborated and cooperated with each other while learning engineering courses in a virtual environment. The students had hands-on experience on their course related labs at their convenient time and location. Apart from that, students also practiced software development activities that might help in their job placements. Some of the lessons learnt in our proposed approach are

- Resources for students should be clear and informative.
- Even though institutes are tiered at the same level, students have different skill sets and the complexity of software developments activities should be accordingly categorized for selection.
- Faculty and lab assistants have to specially trained on collaboration platforms as interactions and assistance required during remote learning are different and more as compared to face-2-face learning.

Some of the possible future directions for this work are :

- Increase the count of students and have students from other engineering or non-engineering disciplines. There may be some constraints on the availability of open source repositories containing simulations or virtual labs of other disciplines.
- Have students develop new labs as well, apart from using and fixing existing code base of the labs.
- Validate learning effectiveness of the approach against conducting lab in a physical and virtual environment.
- Roll-out large scale remote internship so that students and open source community benefits from contributions, a secondary effect of CoCo learning approach.
- Implement Function Point mechanism instead of using Caper Jones mapper on Lines of Code to validate the productivity improvement.
- Validate the survey question responses on conducting labs with the system usage logs.

References

[1] C. R. Graham, "Blended Learning Systems," *The Handbook of Blended Learning*, pp. 3–21, 2006.

[2] A. Moore, *Teaching and Learning: Pedagogy, Curriculum and Culture*. Routledge, 2012.

[3] J. M. McInnerney and T. S. Roberts, "Collaborative and Cooperative Learning," pp. 319–326, 2009.

[4] K. F. Hew and W. S. Cheung, "Students' and Instructors' Use of Massive Open Online Courses (MOOCs): Motivations and Challenges," *Educational research review*, vol. 12, pp. 45–58, 2014.

[5] S. K. L. Riley and K. Stacy, "Teaching in Virtual Worlds: Opportunities and Challenges," *Setting Knowledge Free: The Journal of Issues in Informing Science and Information Technology*, vol. 5, no. 5, pp. 127–135, 2008.

[6] C. Kalina and K. Powell, "Cognitive and Social Constructivism: Developing tools for an Effective Classroom," *Education*, vol. 130, no. 2, pp. 241–250, 2009.

[7] M. Prince, "Does Active Learning Work? A Review of the Research," *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, 2004.

[8] R. E. Slavin, "Developmental and Motivational Perspectives on Cooperative Learning: A Reconciliation," *Child Development*, 1987.

[9] B. L. Smith and J. T. MacGregor, "What is Collaborative Learning?" pp. 233–267, 1992.

[10] D. McConnell, *Implementing Computing Supported Cooperative Learning*. Routledge, 2014.

[11] R. E. Slavin, "Cooperative Learning," *Review of Educational Research*, vol. 50, no. 2, pp. 315–342, 1980.

[12] O. J. Olivares, "Collaborative vs. Cooperative Learning: The Instructor's Role in Computer Supported Collaborative Learning," in *Computer-supported Collaborative Learning: Best Practices and Principles for Instructors*. IGI Global, 2008, pp. 20–39.

[13] G. Stahl, T. D. Koschmann, and D. D. Suthers, *Computer-supported Collaborative Learning*. The Cambridge Handbook of the Learning Sciences, 2014.

[14] C. Huang, "Virtual Labs: e-Learning for Tomorrow," *PLoS Biology*, vol. 2, no. 6, p. e157, 2004.

[15] M. M. Waldrop, "Education online: The Virtual Lab," *Nature News*, vol. 499, no. 7458, p. 268, 2013.

[16] A. Vee, "Understanding Computer Programming as a Literacy," *Literacy in Composition Studies*, vol. 1, no. 2, pp. 42–64, 2013.

[17] T. D. LaToza and A. Van Der Hoek, "Crowdsourcing in Software Engineering: Models, Motivations, and Challenges," *IEEE software*, vol. 33, no. 1, pp. 74–80, 2015.

[18] J. Howe, "The Rise of Crowdsourcing," *Wired Magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[19] B. Fitzgerald and K.-J. Stol, "The Dos and Dont's of Crowdsourcing Software Development," in *International Conference on Current Trends in Theory and Practice of Informatics*. Springer, 2015, pp. 58–64.

[20] K. Mao, L. Capra, M. Harman, and Y. Jia, "A Survey of the Use of Crowdsourcing in Software Engineering," *Journal of Systems and Software*, vol. 126, pp. 57–84, 2017.

[21] A. Cockburn, *Agile Software Development: The Cooperative Game*. Pearson Education, 2006.

[22] E. Thompson, A. Luxton-Reilly, J. L. Whalley, M. Hu, and P. Robbins, "Bloom's Taxonomy for CS Assessment," in *Proceedings of the 10th Conference on Australasian Computing Education*. Australian Computer Society, Inc., 2008, pp. 155–161.

[23] A. Budhiraja, K. Dutta, M. Shrivastava, and R. Reddy, "Towards Word Embeddings for Improved Duplicate Bug Report Retrieval in Software Repositories," in *Proceedings of SIGIR International Conference on Theory of Information Retrieval*. ACM, 2018, pp. 167–170.

[24] C. Jones, "Applied Software Measurement: Global Analysis of Productivity and Quality," 2008.

[25] Y. Zheng, B. Wang, Y. Zheng, and L. Shi, "Estimation of Software Projects Effort based on Function Point," in *4th International Conference on Computer Science and Education*. IEEE, 2009, pp. 941–943.

[26] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality Control in Crowdsourcing Systems: Issues and Directions," *IEEE Internet Computing*, vol. 17, no. 2, 2013.

[27] (2020 (accessed on Jul-30-2020)) Virtual Labs. Available at https://vlab.co.in/.

[28] (2020 (accessed on July-30-2020)) Source code repositories of Virtual Labs. Available at https://github.com/virtual-labs/.

[29] (2020 (accessed on July-30-2020)) Source Code Repository for Identifying Duplicate Logged Issues. Available at https://tinyurl.com/GitDupChk.

[30] (2020 (accessed on July-30-2020)) GitHub CoCo. Available at https://tinyurl.com/GitHubCoCo .